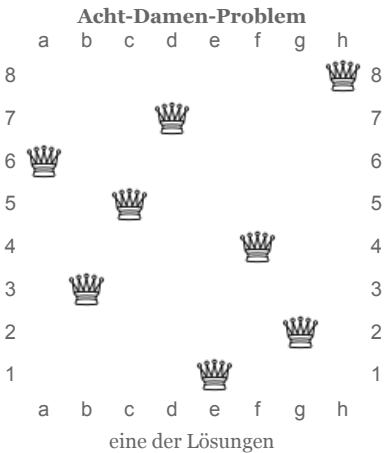


Damenproblem

Das **Damenproblem** ist eine schachmathematische Aufgabe. Es sollen jeweils acht Damen auf einem Schachbrett so aufgestellt werden, dass keine zwei Damen einander gemäß ihren in den Schachregeln definierten Zugmöglichkeiten schlagen können. Die Figurenfarbe wird dabei ignoriert, und es wird angenommen, dass jede Figur jede andere angreifen könnte. Solcherart auf dem Schachbrett angeordnete Figuren werden auch als „unabhängig“ bezeichnet.^[1] Für Damen heißt dies konkret und anders ausgedrückt: Es dürfen keine zwei Damen auf derselben Reihe, Linie oder Diagonale stehen.

Im Mittelpunkt steht beim Damenproblem die Frage nach der Anzahl der möglichen Lösungen. Im Falle des klassischen 8 × 8-Schachbretts gibt es **92** verschiedene Möglichkeiten, die Damen entsprechend aufzustellen. Betrachtet man Lösungen als gleich, die sich durch Spiegelung oder Drehung des Brettes auseinander ergeben, verbleiben noch **12** Basis-Lösungen.

Das Problem kann auf quadratische Schachbretter beliebiger Größe verallgemeinert werden: Dann gilt es, **n** unabhängige Damen auf einem Brett von **n × n** Feldern zu positionieren (mit **n** als Parameter aus den natürlichen Zahlen).



Inhaltsverzeichnis

Geschichte

Anzahl der Lösungen im klassischen Damenproblem

Anzahl der Lösungen im verallgemeinerten Damenproblem

Anzahl der Lösungen bis zur Brettgröße **27 × 27**

Anzahl der Lösungen für große **n**

Algorithmen zur Lösungssuche

Rekursives Backtracking

Constraintprogrammierung

Explizite Lösung

Andere Methoden

Verwandte Problemstellungen

Andere Figuren

Andere Brettgeometrie

Andere Aufgabenstellungen

Siehe auch

Literatur

Weblinks

Einzelnachweise

Geschichte

Erstmals formuliert wurde das Damenproblem von dem bayerischen Schachmeister Max Bezzel. In der Berliner *Schachzeitung* fragte er 1848 nach der Anzahl der möglichen Lösungen. Als erster nannte 1850 der Zahnarzt Franz Nauck in der Leipziger *Illustrierten Zeitung* die korrekte Zahl **92**. 1874 bewies der englische Mathematiker James Whitbread Lee Glaisher, dass es nicht mehr Lösungen geben kann.^[2] Damit war das ursprüngliche Problem vollständig gelöst. Auch Carl Friedrich Gauß zeigte Interesse an dem Problem, weshalb es irrtümlich häufig auf ihn zurückgeführt wird; er hatte indessen nur **72** der Lösungen gefunden.^[3]

Nauck verallgemeinerte die Problemstellung und fragte, auf wie viele verschiedene Arten **n** Damen auf einem **n × n**-Schachbrett aufgestellt werden können.

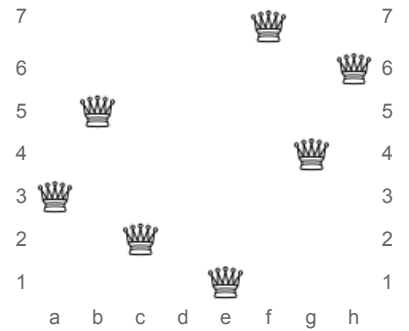
Im Jahre 1992 fanden Demirörs, Rafrak und Tanik eine Äquivalenz zwischen magischen Quadraten und Damenproblemen.^[4]



Das Damenproblem tauchte auch in den Computerspielen The 7th Guest und The Whispered World auf. Im Nintendo DS Titel Professor Layton und das geheimnisvolle Dorf muss es vom Spieler in unterschiedlicher Form sogar mehrfach gelöst werden.

Anzahl der Lösungen im klassischen Damenproblem

Das klassische Problem mit acht Damen auf einem 8×8 -Brett hat **92** verschiedene Lösungen. Wenn man solche, die durch Drehen oder Spiegeln des Brettes aufeinander abgebildet werden, nur einfach zählt, bleiben **12** eindeutige Lösungen übrig (die unterschiedlichen Farben der Felder werden nicht beachtet). Da es für jede dieser reduzierten Lösungen vier Spiegelungen (an Diagonalen, Horizontale und Vertikale durch die Brettmitte) und vier Rotationen gibt, könnte man eine Gesamtzahl von $8 \times 12 = 96$ Lösungen vermuten. Da aber eine der Lösungen (siehe Diagramm) bei einer Drehung um 180° in sich selbst übergeht, lassen sich aus dieser nur vier verschiedene Lösungen konstruieren und es ergeben sich insgesamt **92** Lösungen.



Diese Art sorgt für **92** statt **96** verschiedene Lösungen (basierend auf **12** eindeutigen Lösungen)

Anzahl der Lösungen im verallgemeinerten Damenproblem

Das verallgemeinerte Damenproblem verlangt, n Damen auf einem Brett von $n \times n$ Feldern so zu positionieren, dass sie einander nicht diagonal, senkrecht und waagrecht gegenüber stehen.

Anzahl der Lösungen bis zur Brettgröße 27×27

Die folgende Tabelle führt die Anzahl der eindeutigen Lösungen und die der gesamten Lösungen bis zur Brettgröße 27×27 auf:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
eindeutig	1	0	0	1	2	1	6	12	46	92	341	1.787	9.233	45.752	285.053	1.846.955	11.977.939	83.263.591
insgesamt	1	0	0	2	10	4	40	92	352	724	2.680	14.200	73.712	365.596	2.279.184	14.772.512	95.815.104	666.090.624

n	19	20	21	22	23	24
eindeutig	621.012.754	4.878.666.808	39.333.324.973	336.376.244.042	3.029.242.658.210	28.439.272.956.934
insgesamt	4.968.057.848	39.029.188.884	314.666.222.712	2.691.008.701.644	24.233.937.684.440	227.514.171.973.736

n	25 (errechnet 2005)	26 (errechnet 2009)	27 (errechnet 2016)
eindeutig	275.986.683.743.434	2.789.712.466.510.289	29.363.495.934.315.694
insgesamt	2.207.893.435.808.352	22.317.699.616.364.044	234.907.967.154.122.528

Die zuvor bekannte, aber nicht überprüfte Lösungszahl für $n = 12$ wurde 1969 unabhängig voneinander von Torbjörn Lindelöf und Bernd Schwarzkopf per Computeranalysen bestätigt. Lindelöf errechnete dabei auch die Größen von $n = 13$ und $n = 14$.^[5] 1970 errechnete Lindelöf die Zahl für $n = 15$ mit der Bemerkung, dass Computer die 50- bis 100-fache der zu dieser Zeit üblichen Rechenkapazität für größere Schachbretter benötigen.^[6]

Die Lösungszahl für $n = 26$ wurde am 11. Juli 2009 vom Queens@TUD-Projekt^[7] mit FPGA-Lösern bestimmt und am 30. August 2009 vom MC#-Projekt^[8] auf zwei russischen Superrechnern der damals aktuellen TOP500-Liste bestätigt. Nach mehr als 7 Jahren folgte am 19. September 2016 die Lösungszahl für $n = 27$.^[9] Diese konnte in einem Nachfolgeprojekt^[10] unter Ausnutzung weiterer Problemsymmetrien und der gesteigerten technischen Leistungsfähigkeit wieder mit Hilfe von FPGA-Lösern berechnet werden. Die Bestätigung dieser Zahl durch ein zweites unabhängiges Projekt steht noch aus.

Allgemein lässt sich feststellen, dass die Anzahl der Lösungen etwas schneller als exponentiell mit der Brettgröße anwächst. Interessanterweise gibt es auf dem 2×2 -Brett sowie auf dem 6×6 -Brett weniger Lösungen als auf dem jeweils kleineren Brett.

Anzahl der Lösungen für große n

Eine obere Schranke für die Lösungsanzahl $D(n)$ des Damenproblems auf einem $n \times n$ -Brett ist $n!$. Dies ist die Anzahl von Lösungen für n einander nicht bedrohende Türme. Die Aufstellungen voneinander nicht bedrohenden Damen (für $n > 1$) sind eine echte Teilmenge hiervon.

Die asymptotische Form von $D(n)$ ist nicht bekannt. Rivin u. a. stellen die Vermutung auf, dass

$$\lim_{n \rightarrow \infty} \frac{\log D(n)}{n \log n} = \beta > 0 \text{ ist.}^{[11]}$$

Aus den bekannten Gliedern der Folge lässt sich für große n folgende Näherungsformel abschätzen:^[12]

$$D(n) \approx c^{n-1} \times n! \text{ mit } c = \frac{243}{625} \approx 0,3888.$$

Algorithmen zur Lösungssuche

Das Damenproblem ist ein gutes Beispiel für ein einfach zu formulierendes Problem mit nicht-trivialen Lösungen. Eine Reihe von Programmiertechniken ist geeignet, alle Lösungen zu erzeugen. Klassisch ist rekursives Backtracking; dieses ist besonders einfach zu realisieren mit logischer Programmierung. Eine weitere Möglichkeit sind genetische Algorithmen.

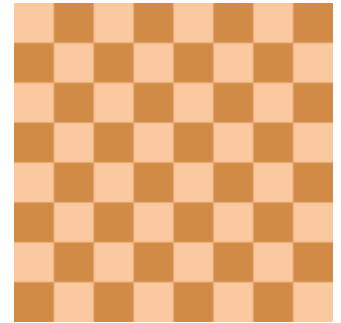
Derartige Ansätze sind wesentlich effizienter als ein naiver Brute-Force-Algorithmus, der (im 8×8 -Fall) alle $64 \times 63 \times 62 \times 61 \times 60 \times 59 \times 58 \times 57$ ($< 64^8 = 2^{48}$) möglichen Positionierungen der acht Damen durchprobiert und dabei alle Stellungen ausfiltert, in denen jeweils zwei Damen einander schlagen könnten. Dieser Algorithmus erzeugt mehrfach die gleichen Lösungen, wenn Permutationen der Damen gleiche Felder besetzen.

Ein etwas effizienterer Brute-Force-Algorithmus platziert nacheinander in jeder Reihe nur eine Dame und reduziert dadurch die Komplexität auf $8^8 = 2^{24}$ mögliche Stellungen.

Rekursives Backtracking

Das Damenproblem kann durch einen rekursiven Algorithmus effizient gelöst werden, indem das Problem mit n Damen so aufgefasst wird, dass es gilt, zu jeder Lösung mit $n - 1$ Damen eine weitere Dame hinzuzufügen. Letztendlich lässt sich jedes Damenproblem damit auf ein Problem mit 0 Damen zurückführen, was nichts anderes als ein leeres Schachbrett ist.

Das folgende Python-Programm findet alle Lösungen des n -Damen-Problems mit Hilfe eines rekursiven Algorithmus. Ein $n \times n$ -Brett wird dabei rekursiv auf kleinere Bretter mit geringerer Anzahl an Reihen, $n \times (n - 1)$, $n \times (n - 2)$, ... reduziert. Das Programm nutzt direkt aus, dass keine zwei Damen in der gleichen Reihe stehen. Außerdem wird benutzt, dass eine Lösung mit $n - 1$ Damen auf einem $n \times (n - 1)$ -Brett auf jeden Fall eine Lösung mit $n - 2$ Damen auf einem $n \times (n - 2)$ -Brett enthalten muss. In anderen Worten, wenn man die untere (oder obere) Reihe der Teillösung eines $n \times (n - 1)$ -Bretts entfernt, bleiben $n - 2$ Damen auf einem $n \times (n - 2)$ -Brett übrig, die wiederum eine Teillösung auf dem $n \times (n - 2)$ -Brett darstellen.



Animation des rekursiven Backtracking-Algorithmus

Der Algorithmus konstruiert also alle Lösungen aus den Lösungen eines jeweils kleineren Brettes. Da sichergestellt wird, dass die Teillösungen auf den kleinen Brettern konfliktfrei sind, spart dieser Algorithmus das Überprüfen vieler Stellungen. Insgesamt werden für das 8×8 -Brett nur **15.720** Stellungen überprüft.

```
# Erzeuge eine Liste von Lösungen auf einem Brett mit Reihen und Spalten.
# Eine Lösung wird durch eine Liste der Spaltenpositionen,
# indiziert durch die Reihenummer, angegeben.
# Die Indizes beginnen mit Null.
def damenproblem(reihen, spalten):
    if reihen <= 0:
        return [[]] # keine Dame zu setzen; Leeres Brett ist Lösung
    else:
        return eine_dame_dazu(reihen - 1, spalten, damenproblem(reihen - 1, spalten))

# Probiere alle Spalten, in denen für eine gegebene Teillösung
# eine Dame in "neue_reihe" gestellt werden kann.
# Wenn kein Konflikt mit der Teillösung auftritt,
# ist eine neue Lösung des um eine Reihe erweiterten
# Bretts gefunden.
def eine_dame_dazu(neue_reihe, spalten, vorherige_loesungen):
    neue_loesungen = []
    for loesung in vorherige_loesungen:
        # Versuche, eine Dame in jeder Spalte von neue_reihe einzufügen.
        for neue_spalte in range(spalten):
            # print('Versuch: %s in Reihe %s' % (neue_spalte, neue_reihe))
            if kein_konflikt(neue_reihe, neue_spalte, loesung):
                # Kein Konflikte, also ist dieser Versuch eine Lösung.
                neue_loesungen.append(loesung + [neue_spalte])
    return neue_loesungen

# Kann eine Dame an die Position "neue_spalte"/"neue_reihe" gestellt werden,
# ohne dass sie eine der schon stehenden Damen schlagen kann?
def kein_konflikt(neue_reihe, neue_spalte, loesung):
    # Stelle sicher, dass die neue Dame mit keiner der existierenden
    # Damen auf einer Spalte oder Diagonalen steht.
    for reihe in range(neue_reihe):
        if (loesung[reihe] == neue_spalte or
```

```

    loesung[reihe] + reihe == neue_spalte + neue_reihe or # gleiche Diagonale
    loesung[reihe] - reihe == neue_spalte - neue_reihe): # gleiche Diagonale
        return False
    return True

for loesung in damenproblem(8, 8):
    print(loesung)

```

Durch die Verwendung einer Koroutine kann der Algorithmus etwas vereinfacht werden. Das folgende Programm ist eine Übersetzung der Lösung von Niklaus Wirth in die Programmiersprache Python, verzichtet jedoch auf die im Original verwendete Indexarithmetik und verwendet stattdessen einfache Listen. An die Stelle der Prozedur tritt hier ein Generator: eine eingeschränkte Form einer Koroutine, durch die ein Iterator erzeugt wird.^[13]

```

def tryout(n, i, a, b, c):
    if i < n:
        for j in range(n):
            if j not in a and i+j not in b and i-j not in c:
                for solution in tryout(n, i+1, a+[j], b+[i+j], c+[i-j]):
                    yield solution
        else:
            yield a
    for solution in tryout(8, 0, [], [], []):
        print(solution)

```

Constraintprogrammierung

Die Constraintprogrammierung über endliche Bereiche kann eine Aufgabe wie das Damenproblem sehr kompakt formulieren. Das folgende Prolog-Programm (in *GNU Prolog*) findet schnell eine Lösung auch für große Schachbretter.

```

/* Dieses Prädikat erzeugt eine Liste, die eine einzige Lösung darstellt.
   Es ist sichergestellt, dass jeder Wert zwischen 1 und N genau einmal auftritt. */

n_damen(N,Ls) :- length(Ls,N),
                 fd_domain(Ls,1,N),
                 fd_all_different(Ls),
                 constraint_damen(Ls),
                 fd_labeling(Ls,[variable_method(random)]).

/* Dieses Prädikat stellt sicher,
   dass alle Stellungen die Lösungsbedingungen erfüllen */

constraint_damen([]).
constraint_damen([X|Xs]) :- nicht_schlagen(X,Xs,1), constraint_damen(Xs).

/* Mit diesem Prädikat wird sichergestellt,
   dass zwei Damen nicht auf einer Diagonalen stehen */

nicht_schlagen(_,[],_).
nicht_schlagen(X,[Y|Xs],N) :- X#\=Y+N, X#\=Y-N, T#=-N+1, nicht_schlagen(X,Xs,T).

```

Explizite Lösung

Eine Konstruktionsvorschrift für eine spezielle Lösung mit beliebig großem n wurde erstmals 1874 von Emil Pauls angegeben.^{[14][15][16]} Hierdurch wurde also insbesondere bewiesen, dass das Damenproblem für beliebiges $n > 3$ mindestens eine Lösung besitzt.

Für gerade n , die bei Division mit **6** den Rest **0** oder **4** ergeben, starte man in der zweiten Spalte der obersten Zeile und platziere eine Dame. Platziere die folgenden Damen jeweils zwei Spalten rechts und eine Zeile unter der vorigen, bis $\frac{n}{2}$ Zeilen gefüllt sind. Die Zeilen der unteren Brethälfte erhält man aus der Spiegelung der oberen Damen am Mittelpunkt des Bretts.

Für gerade n , die bei Division mit **6** den Rest **2** ergeben (darunter das normale Schachbrett mit $n = 8$) führt diese Vorschrift nicht zu einer gültigen Lösung. Für diesen Fall lässt sich eine alternative, etwas komplizierte Konstruktionsvorschrift angeben.^{[16][17]}

Bei beiden Konstruktionsvorschriften bleiben alle Felder der langen Diagonale (links oben nach rechts unten) unbesetzt. Für ungerade n bilde man daher nach obigen Vorschriften eine gültige Lösung für $n - 1$ und platziere die letzte Dame in die letzte Spalte der letzten Zeile.

	a	b	c	d	e	f	
6		♔					6
5				♔			5
4						♔	4
3	♔						3
2			♔				2
1					♔		1
	a	b	c	d	e	f	

Konstruktionsvorschrift für gerades n , das bei Division durch **6** nicht den Rest **2** ergibt

Andere Methoden

Ein *iterativer Reparaturalgorithmus* beginnt mit einer beliebigen Stellung der Damen auf dem Brett. Es zählt dann die Anzahl der Konflikte, und versucht, durch Umpositionieren der Damen die Anzahl der Konflikte zu reduzieren. Effizient ist etwa, die Dame mit den meisten Konflikten senkrecht auf die Position zu verschieben, auf der die geringste Anzahl von Konflikten auftritt. Mit dieser Methode kann das **1.000.000-Damen-Problem** ausgehend von einer „vernünftigen“ Versuchsposition gelöst werden. Derart große Bretter lassen sich mit expliziten Konstruktionsalgorithmen nicht lösen (triviale Lösungen ausgenommen); allerdings kann ein solcher Iterationsalgorithmus nicht mit Sicherheit eine Lösung finden.

Verwandte Problemstellungen

Andere Figuren

Das Problem kann auch für andere Schachfiguren (König, Läufer, Springer, Turm) formuliert werden.

Eine weitere Verallgemeinerung des Problems stellt das ***n**-Superdamen-Problem* dar. Superdamen dürfen wie Damen und Springer ziehen. Es ist nicht klar, wer Superdamen oder das ***n**-Superdamen-Problem* erfunden hat. In *Mathematische Knocheleien* (Vieweg-Verlag, 1973) erwähnt Martin Gardner eine Schachvariation, in der mit einer Superdame gespielt wird. Gardner nennt diese Figur dort *Maharadscha*. Andere kennen sie als Amazone.

Auch die Frage, auf wie viele Arten sich ***n*** Superdamen auf einem ***n** × **n***-Schachbrett bedrohungsfrei platzieren lassen, wurde untersucht.^[18]

n	Lösungen	n	Lösungen	n	Lösungen	n	Lösungen	n	Lösungen	n	Lösungen
1	1	6	0	11	44	16	202.900	21	3.977.841.852	26	286.022.102.245.804
2	0	7	0	12	156	17	1.330.622	22	34.092.182.276		
3	0	8	0	13	1.876	18	8.924.976	23	306.819.842.212		
4	0	9	0	14	5.180	19	64.492.432	24	2.883.202.816.808		
5	0	10	4	15	32.516	20	495.864.256	25	28.144.109.776.812		

Seit 2001 existiert auch für das ***n**-Superdamen-Problem* eine explizite Lösung von Frank Schwelling.

Angemerkt sei, dass das Springerproblem nicht die analoge Aufgabe für Springer ist, sondern eine Springerwanderung über das ganze Schachbrett.

Andere Brettgeometrie

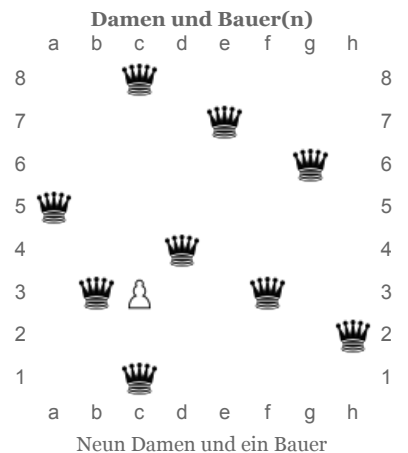
George Pólya betrachtete das Damenproblem auf einem torusförmigen Brett. Er bewies, dass genau dann mindestens eine Lösung existiert, wenn ***n*** zu **6** teilerfremd ist, also weder durch **2** noch durch **3** teilbar ist. Auch dreidimensionale Verallgemeinerungen wurden untersucht^[19].

Andere Aufgabenstellungen

Für ein ***n** × **n***-Schachbrett bestimme man die Dominanzzahl, das ist die Mindestzahl an Damen, die ausreicht, jedes Feld des Brettes zu beherrschen. Auf dem **8 × 8**-Brett reichen fünf Damen aus. Dafür gibt es **4.860** Lösungen (etwa b7, d5, e4, f3, h1).^[20]

Das Neun-Damen-Problem verlangt, auf einem **8 × 8**-Brett neun Damen und einen Bauern derart unterzubringen, dass die Damen einander nicht beobachten können, also keine direkte waagerechte, senkrechte oder diagonale Sichtlinie zueinander haben. Dieses Problem kann wiederum auf beliebige Brettgröße und eine höhere Anzahl von Bauern verallgemeinert werden.

Bei einer Abwandlung des klassischen Damenproblems stehen bereits einige Damen auf dem Brett. Diese sind nicht versetzbar. Gibt es unter diesen Bedingungen eine Lösung des Damenproblems? Für diese Problemstellung haben Mathematiker des Clay Mathematics Institutes 1 Million Dollar für denjenigen ausgelobt, der einen Lösungsalgorithmus findet. Prof. Ian Gent (University of St Andrews) formuliert die Problemstellung: *Wenn einige Damen bereits auf einem n-zu-n-Schachbrett gesetzt sind, kannst du das n-Damenproblem dann lösen, ohne die gesetzten zu bewegen?*^[21]



Siehe auch

- Schachkomposition

Literatur

- John J. Watkins: *Across the Board: The Mathematics of Chess Problems*. Princeton University Press, Princeton 2004, ISBN 0-691-11503-6.

Weblinks

Wiktionary: Damenproblem – Bedeutungserklärungen, Wortherkunft, Synonyme, Übersetzungen

- Eric W. Weisstein: Queens Problem (<http://mathworld.wolfram.com/QueensProblem.html>), MathWorld (englisch)
- Allgemeine Methode n Königinnen mit Implementierung in Java (<https://problema-de-las-1000-reinas-resuelto.blogspot.com/2019/08/metodo-general-8-reinas.html>), (spanisch)
- Berechnung der Lösungen für verschiedene Brettgrößen in JavaScript (http://www.arstechnica.de/computer/JavaScript/JS11_04.html) auf ArsTechnica.de
- Walter Kusters: n -Queens bibliography (<http://liacs.leidenuniv.nl/~kusterswa/nqueens/>) (Literaturliste), Universiteit Leiden (englisch)
- Lösungen in mehr als 100 verschiedenen Programmiersprachen: N -queens problem (http://rosettacode.org/wiki/N-queens_problem) auf rosettacode.org
- Jeff Somers N -Queen-Quellcode (https://web.archive.org/web/20070125091935/http://www.jsomers.com/nqueen_demo/nqueens.html) (englisch; sehr gut beschrieben)
- Takakens N -Queen-Quellcode. (<http://www.ic-net.or.jp/home/takaken/e/queen/index.html>) Ca. 4-mal schneller als Jeff Somers Code

Einzelnachweise

- Jewgeni Gik: *Schach und Mathematik*. Reinhard Becker Verlag, 1986, ISBN 3-930640-37-6.
- Evgeni J. Gik: *Schach und Mathematik*. Verlag Deutsch Harri GmbH, ISBN 3-87144-987-3.
- Klaus Menzel (Hrsg.): *Algorithmen: Vom Problem zum Programm*. Springer-Verlag, 2013, ISBN 978-3-322-91373-9, S. 106 (128 S., eingeschränkte Vorschau (<https://books.google.de/books?id=qpumBgAAQBAJ&pg=PA106&q=%22Carl+Friedrich+Gauss%22+Dame+nproblem+72#v=onepage>) in der Google-Buchsuche).
- Onur Demirörs, Nader Raftaf, Murat Tanik: *Obtaining n -queens solutions from magic squares and constructing magic squares from n -queens Solutions*. In: *Journal of Recreational Mathematics*. Nr. 24, 1992, ISSN 0022-412X (<https://zdb-katalog.de/list.xhtml?t=iss%3D%220022-412X%22&key=cql>), S. 272–280.
- Karl Fabel: *Das n -Damen-Problem*. In: *Die Schwalbe*. Heft 1, Oktober 1969, S. 20.
- Karl Fabel: *Das n -Damen-Problem*. In: *Die Schwalbe*. Heft 5, September 1970, S. 146.
- 26-Damen-Problem gelöst* (<https://www.heise.de/newsticker/meldung/26-Damen-Problem-geloest-6855.html>) auf heise.de, 19. Juli 2009
- MC#-Projekt (<https://web.archive.org/web/20120301124806/http://www.mcsharp.net/nqueens.html>) (Memento vom 1. März 2012 im Internet Archive)
- heise online: *Zahlen, bitte! Das 27-Damen-Problem ist gelöst*. (<https://www.heise.de/newsticker/meldung/Zahlen-bitte-Das-27-Damen-Problem-ist-geloest-3332513.html>) In: *heise online*. Abgerufen am 27. September 2016.
- preusser/q27*. (<https://github.com/preusser/q27>) In: *GitHub*. Abgerufen am 20. September 2016.
- I. Rivin, I. Vardi, P. Zimmermann: *The n -Queens Problem*. In: *The American Mathematical Monthly*. Band 101, 1994, S. 629–639.
- Folge A000170 in OEIS
- Niklaus Wirth: *Algorithmen und Datenstrukturen*, Oberon-Version 2004, Korrektur 2012, S. 114–118. (online (<https://www.inf.ethz.ch/personal/wirth/AD.pdf>); PDF)
- Emil Pauls: *Das Maximalproblem der Damen auf dem Schachbrette I*. Studie aus dem Gebiete des mathematischen Schachs. In: *Deutsche Schachzeitung*. 29. Jahrgang, Nr. 5. Leipzig Mai 1874, S. 129–134 (Digitalisat (<https://books.google.de/books?id=YORgAAAcAAJ&pg=PA129#v=onepage>) in der Google-Buchsuche).
- Emil Pauls: *Das Maximalproblem der Damen auf dem Schachbrette (II. Schluss)*. Studie aus dem Gebiete des mathematischen Schachs. In: *Deutsche Schachzeitung*. 29. Jahrgang, Nr. 9. Leipzig September 1874, S. 257–267 (Digitalisat (<https://books.google.de/books?id=YORgAAAAcAAJ&pg=PA257#v=onepage>) in der Google-Buchsuche).
- W. Ahrens: *Mathematische Unterhaltungen und Spiele Band 1*. Leipzig, 1901, S. 114–156. Digitalisat (<http://gallica.bnf.fr/ark:/12148/bpt6k99446d>)
- Hoffman u. a.: *Construction for the Solutions of the m Queens Problem*. In: *Mathematics Magazine*, Band XX, 1969, S. 66–72. PDF (<http://penguin.ewu.edu/~trolfe/QueenLasVegas/Hoffman.pdf>)
- Folge A051223 in OEIS
- Martin S. Pearson: *Queens On A Chessboard – Beyond The 2nd Dimension* (<http://queens.lyndenlea.info/beyond2d.php>) (en, php) Abgerufen am 27. Januar 2020.
- Alfred Diel: *Das Spiel der Könige*. Bamberger Schachverlag 1983, ISBN 3-923113-03-X, S. 114.
- Eine Million Dollar für denjenigen, der dieses Schachrätsel lösen kann (<https://www.welt.de/kmpkt/article168402274/Eine-Million-Dollar-fuer-denjenigen-der-dieses-Schachraetsel-loesen-kann.html>) veröffentlicht am 11. September 2017 auf welt.de (abgerufen am 6. Dezember 2020)

Abgerufen von „<https://de.wikipedia.org/w/index.php?title=Damenproblem&oldid=207378417>“

Diese Seite wurde zuletzt am 7. Januar 2021 um 23:12 Uhr bearbeitet.

Der Text ist unter der Lizenz „Creative Commons Attribution/Share Alike“ verfügbar; Informationen zu den Urhebern und zum Lizenzstatus eingebundener Mediendateien (etwa Bilder oder Videos) können im Regelfall durch Anklicken dieser abgerufen werden. Möglicherweise unterliegen die Inhalte jeweils zusätzlichen Bedingungen. Durch die Nutzung dieser Website erklären Sie sich mit den Nutzungsbedingungen und der Datenschutzrichtlinie einverstanden. Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.